

data_analysis copy

July 17, 2025

```
[16]: import pandas as pd
      from sklearn.model_selection import train_test_split
      import joblib
      import numpy as np
      from sklearn.feature_selection import VarianceThreshold, SelectKBest, f_classif
```

```
[17]: # Show all columns in the DataFrame
      pd.set_option('display.max_columns', None)
      pd.set_option('display.width', None)
```

```
[18]: df = pd.read_csv('/home/ics-security/ICS-Detection/results/combined/
      ↪Bagged_Ensemble_predictions.csv')
```

```
[19]: df.head()
```

```
[19]:  feature_0  feature_1  feature_2  feature_3  feature_4  feature_5  \
0         12.0         0.0         2.0         0.0         0.0         0.0
1         12.0         0.0         2.0         0.0         0.0         0.0
2          9.0         0.0         2.0         0.0         0.0         0.0
3         29.0         0.0         2.0         0.0         0.0         0.0
4         50.0         0.0         2.0         0.0         0.0         0.0

      feature_6  feature_7  feature_8  feature_9  feature_10  feature_11  \
0          0.0         0.0         0.0         0.0         0.0         0.0
1          0.0         0.0         0.0         0.0         0.0         0.0
2          0.0         0.0         0.0         0.0         0.0         0.0
3          0.0         0.0         0.0         0.0         0.0         0.0
4          0.0         0.0         0.0         0.0         0.0         0.0

      feature_12  feature_13  feature_14  feature_15  feature_16  feature_17  \
0  166666.666667         12.0         0.0         12.0         12.0         0.0
1  166666.666667         12.0         0.0         12.0         12.0         0.0
2  222222.222222          9.0         0.0          9.0          9.0         0.0
3   68965.517241         29.0         0.0         29.0         29.0         0.0
4   40000.000000         50.0         0.0         50.0         50.0         0.0

      feature_18  feature_19  feature_20  feature_21  feature_22  feature_23  \
0          0.0         0.0         0.0         0.0         12.0         12.0
```

1	0.0	0.0	0.0	0.0	12.0	12.0
2	0.0	0.0	0.0	0.0	9.0	9.0
3	0.0	0.0	0.0	0.0	29.0	29.0
4	0.0	0.0	0.0	0.0	50.0	50.0

	feature_24	feature_25	feature_26	feature_27	feature_28	feature_29	\
0	0.0	12.0	12.0	0.0	64.0	0.0	
1	0.0	12.0	12.0	0.0	64.0	0.0	
2	0.0	9.0	9.0	0.0	64.0	0.0	
3	0.0	29.0	29.0	0.0	64.0	0.0	
4	0.0	50.0	50.0	0.0	64.0	0.0	

	feature_30	feature_31	feature_32	feature_33	feature_34	feature_35	\
0	166666.666667	0.0	0.0	0.0	0.0	1.0	
1	166666.666667	0.0	0.0	0.0	0.0	1.0	
2	222222.222222	0.0	0.0	0.0	0.0	1.0	
3	68965.517241	0.0	0.0	0.0	0.0	1.0	
4	40000.000000	0.0	0.0	0.0	0.0	1.0	

	feature_36	feature_37	feature_38	feature_39	feature_40	feature_41	\
0	0.0	0.0	0.0	0.0	0.0	2.0	
1	0.0	0.0	0.0	0.0	0.0	2.0	
2	0.0	0.0	0.0	0.0	0.0	2.0	
3	0.0	0.0	0.0	0.0	0.0	2.0	
4	0.0	0.0	0.0	0.0	0.0	2.0	

	feature_42	feature_43	feature_44	feature_45	feature_46	feature_47	\
0	0.0	83.0	0.0	0.0	0.0	0.0	
1	0.0	83.0	0.0	0.0	0.0	0.0	
2	0.0	83.0	0.0	0.0	0.0	0.0	
3	0.0	83.0	0.0	0.0	0.0	0.0	
4	0.0	83.0	0.0	0.0	0.0	0.0	

	feature_48	feature_49	cm_label
0	0.0	0.0	TN
1	0.0	0.0	TN
2	0.0	0.0	TN
3	0.0	0.0	TN
4	0.0	0.0	TP

```
[20]: # Tạo từ điển ánh xạ từ cm_label sang tên dễ hiểu hơn
label_mapping = {
    'TP': 'Phát hiện đúng MITM',
    'TN': 'Phát hiện đúng bình thường',
    'FP': 'Báo nhầm là MITM',
    'FN': 'Bỏ sót tấn công'
}
```

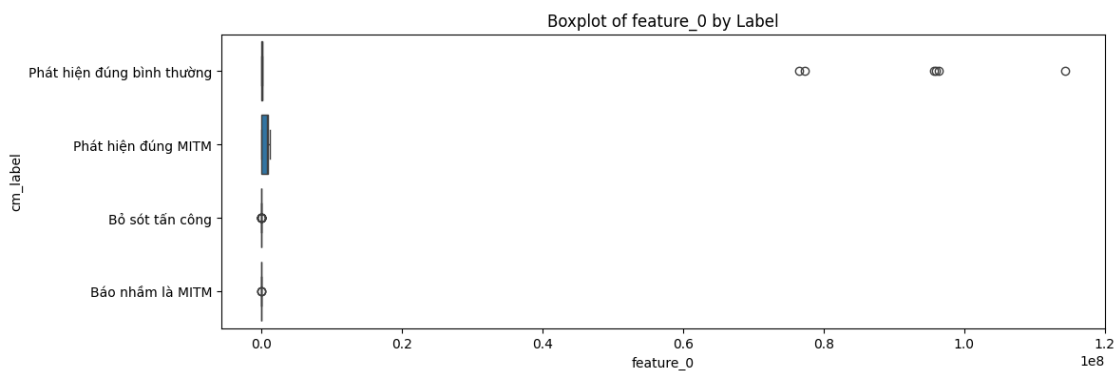
```
# Áp dụng thay thế
df['cm_label'] = df['cm_label'].replace(label_mapping)
```

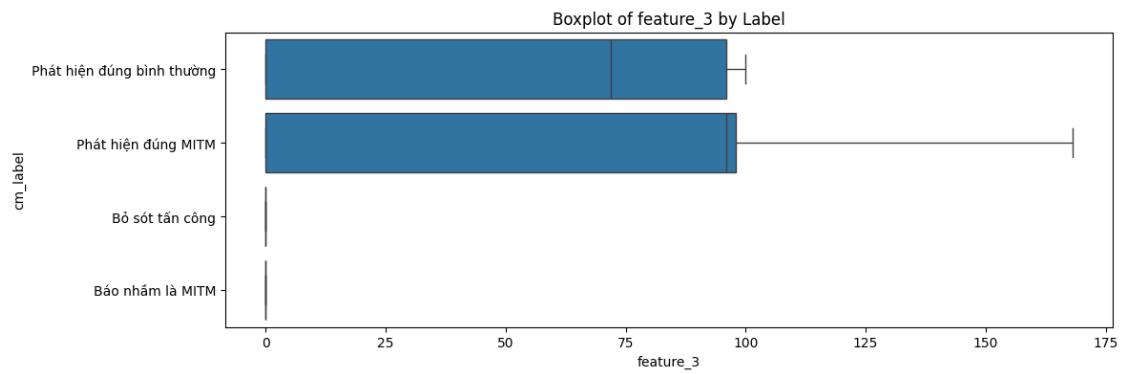
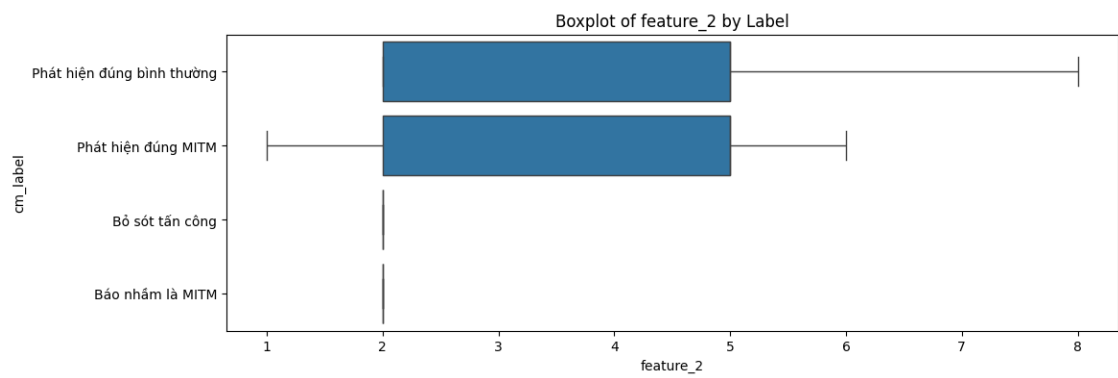
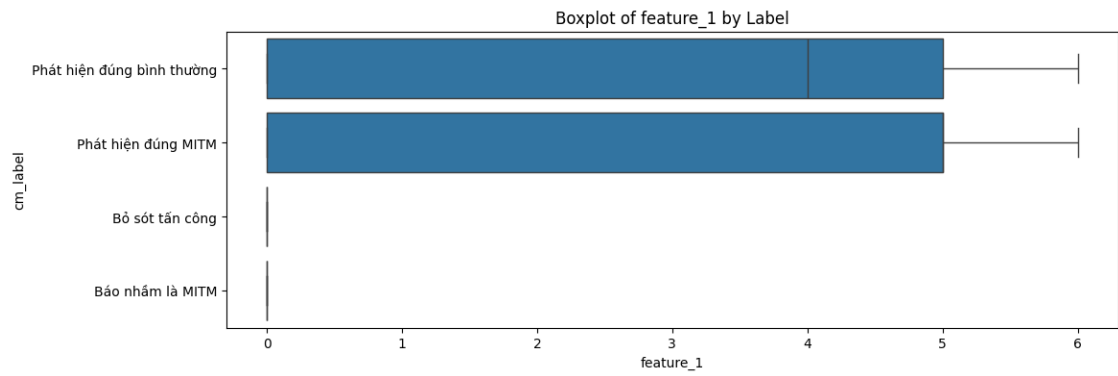
```
[21]: label_counts = df['cm_label'].value_counts()
label_percent = df['cm_label'].value_counts(normalize=True) * 100
result = pd.DataFrame({'Count': label_counts, 'Percentage (%)': label_percent.
    ↳round(2)})
print(result)
```

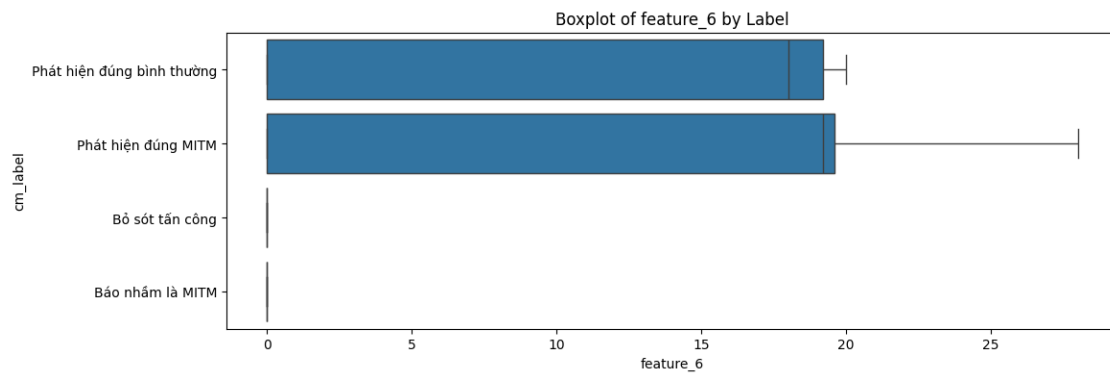
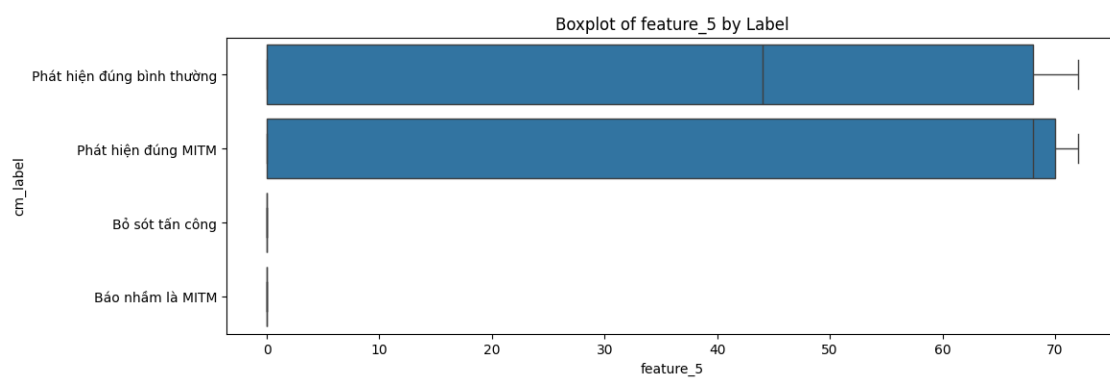
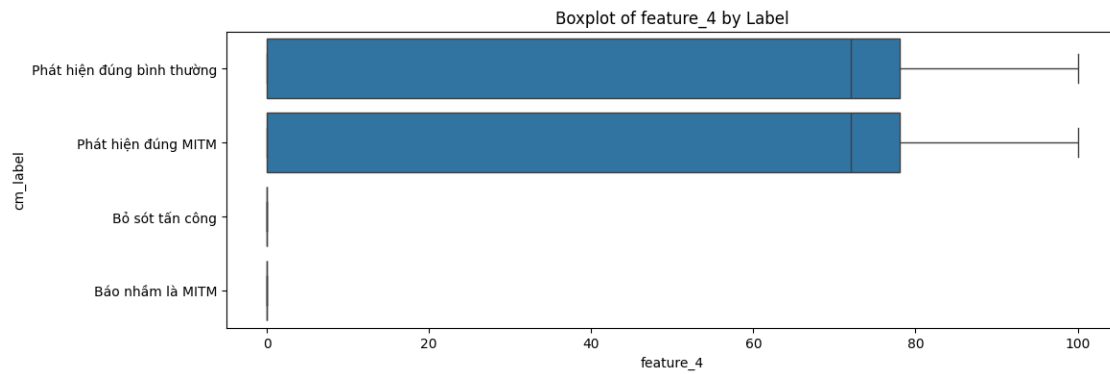
	Count	Percentage (%)
cm_label		
Phát hiện đúng bình thường	24584	88.48
Phát hiện đúng MITM	2376	8.55
Bỏ sót tấn công	741	2.67
Báo nhầm là MITM	85	0.31

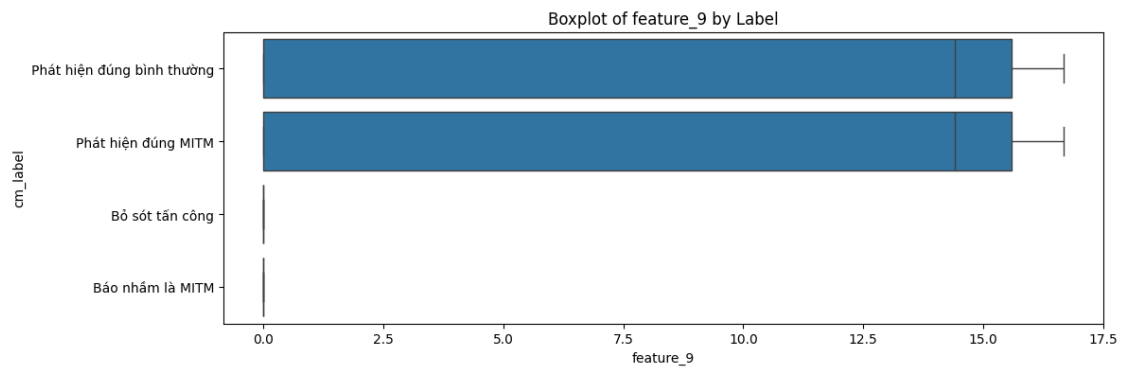
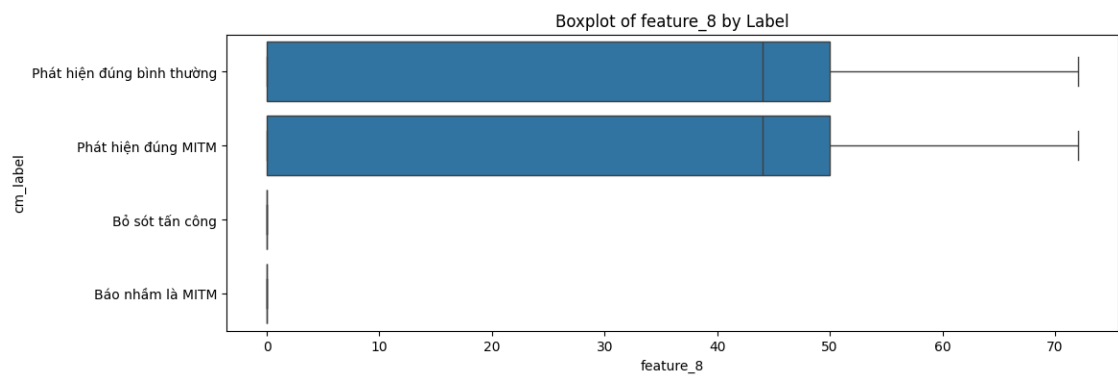
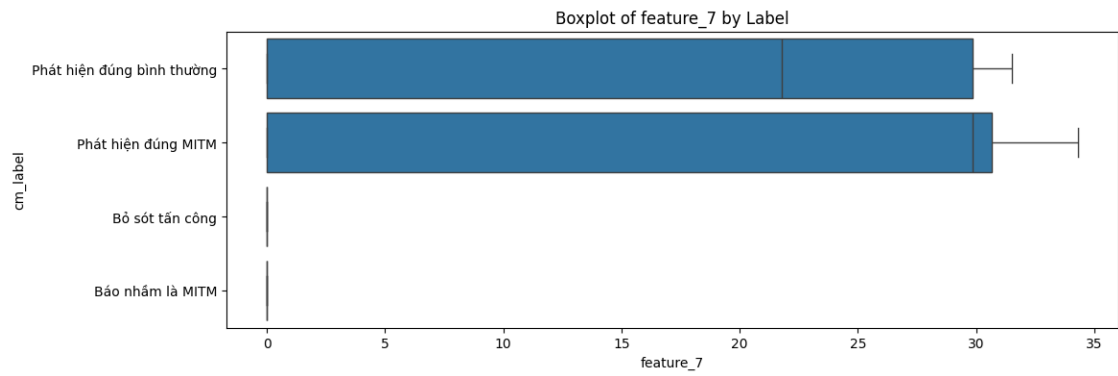
```
[22]: import seaborn as sns
import matplotlib.pyplot as plt

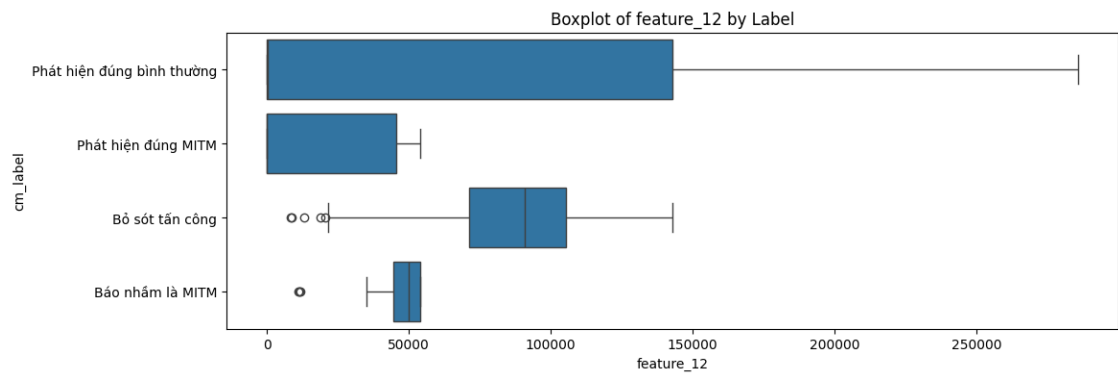
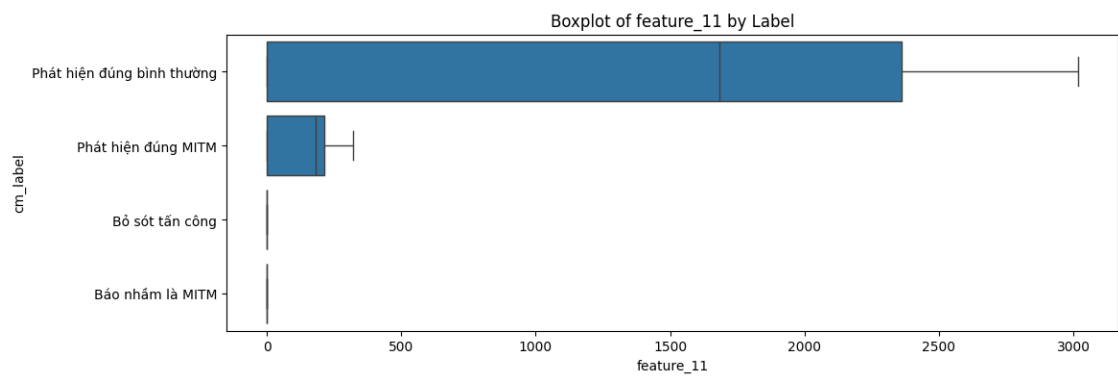
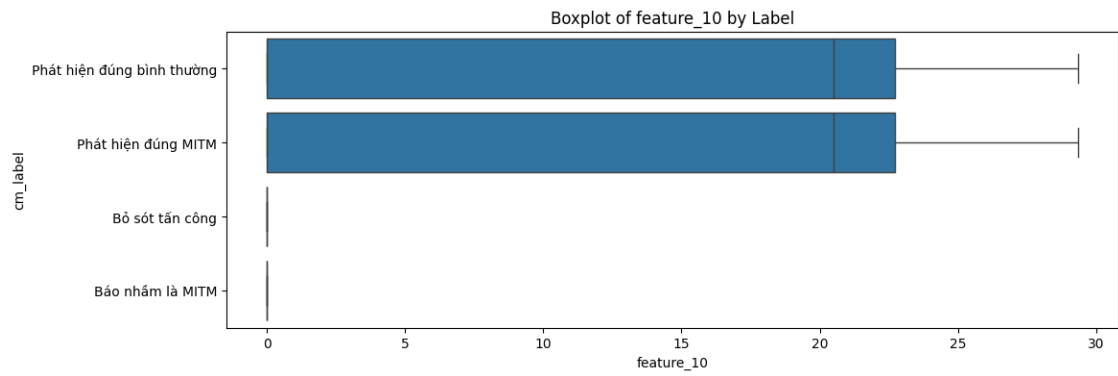
for col in df.select_dtypes(include='number').columns:
    plt.figure(figsize=(12, 4))
    sns.boxplot(y='cm_label', x=col, data=df)
    plt.title(f'Boxplot of {col} by Label')
    plt.show()
```

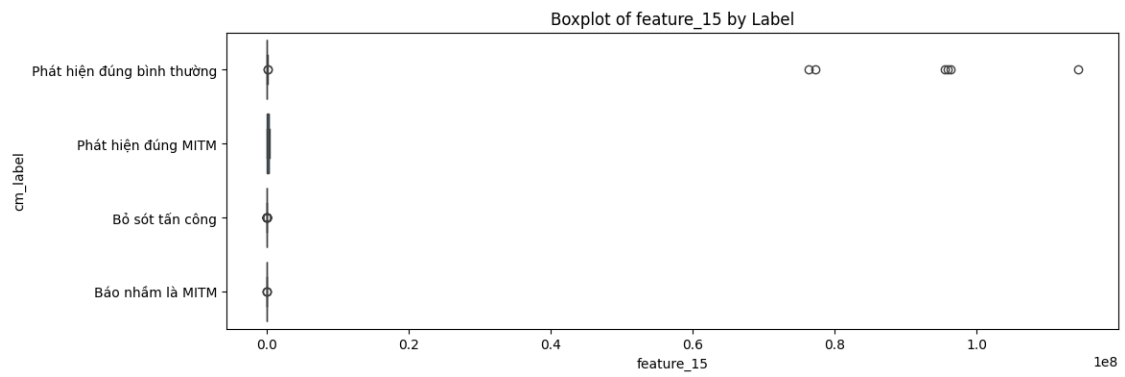
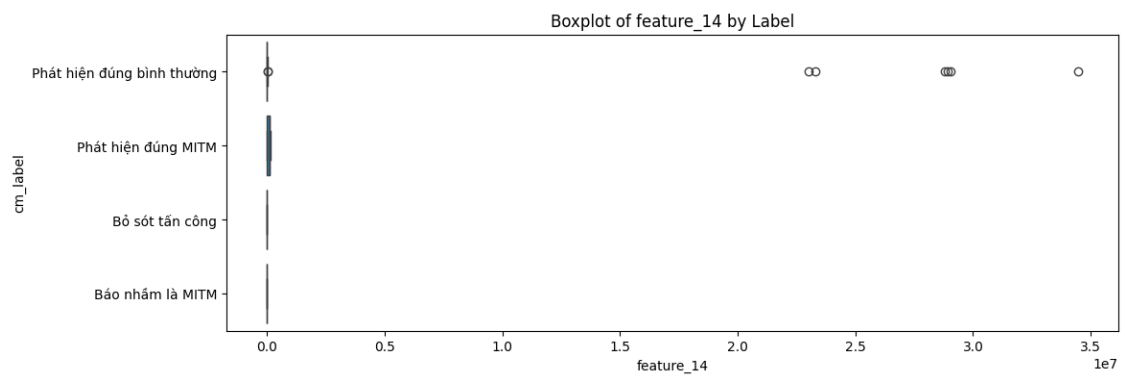
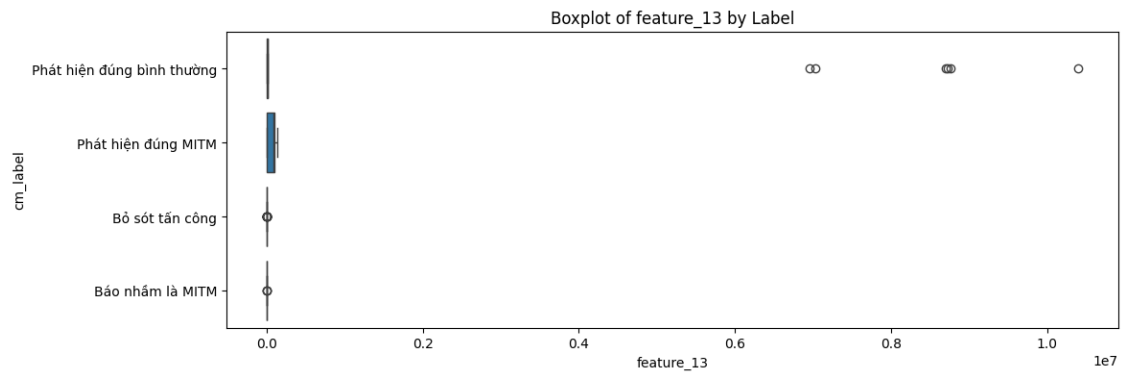


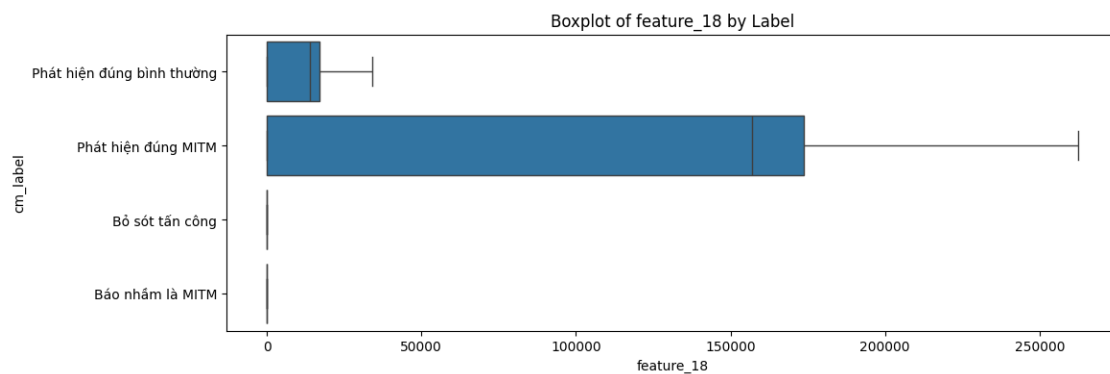
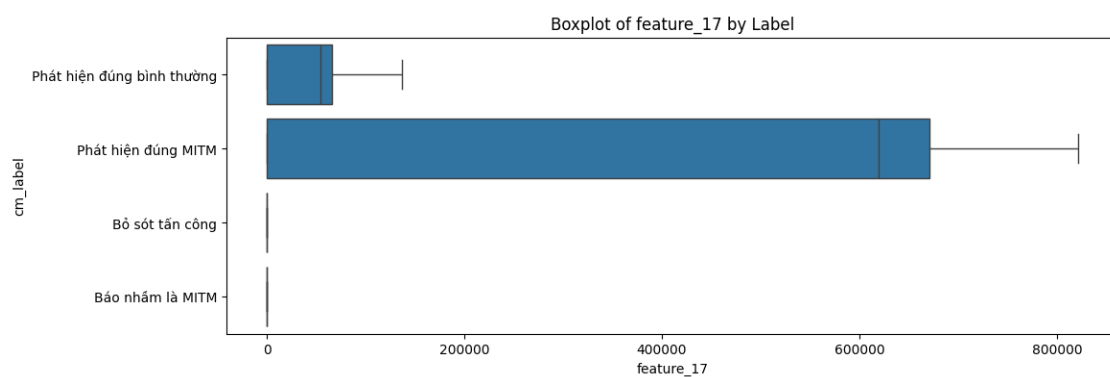
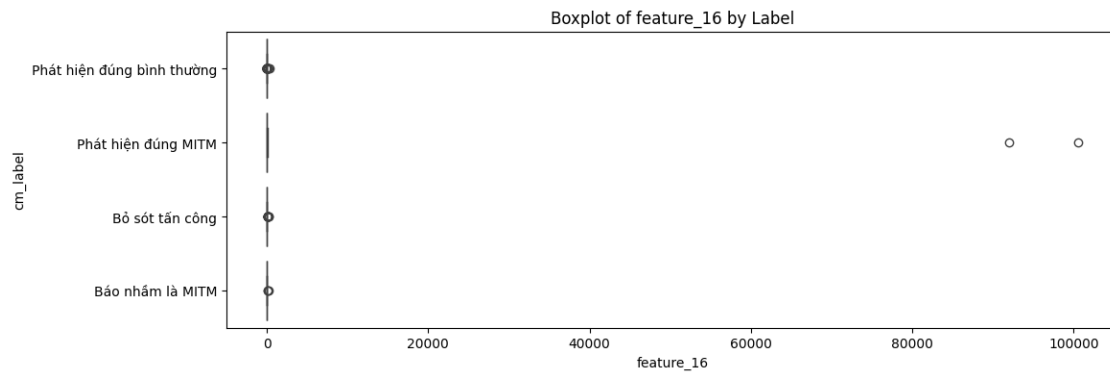


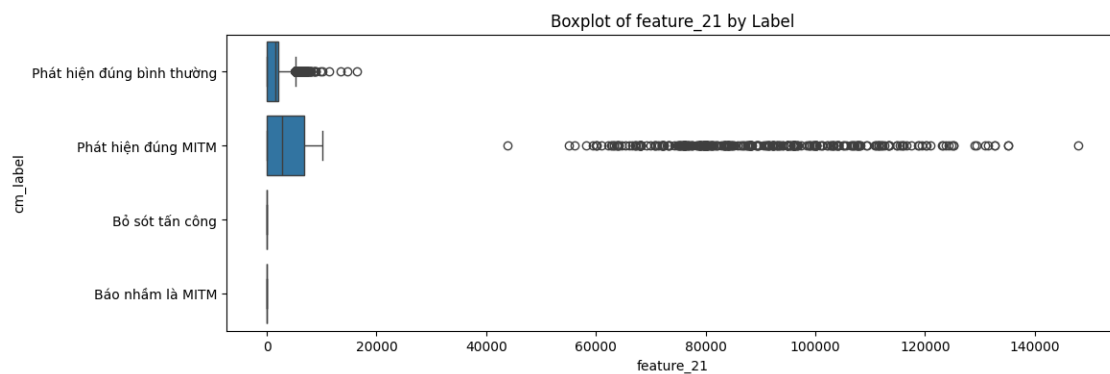
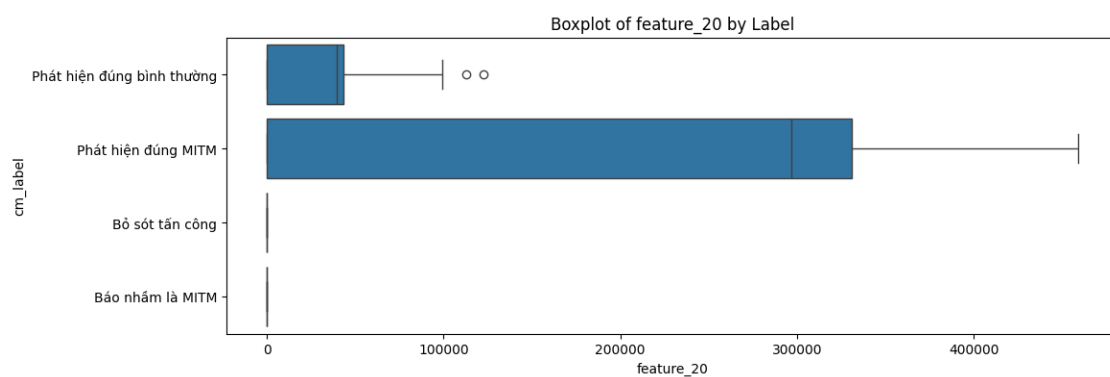
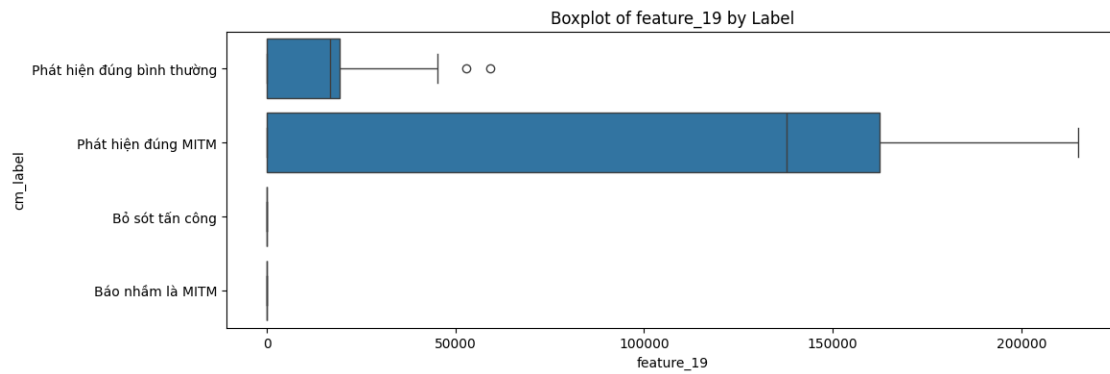


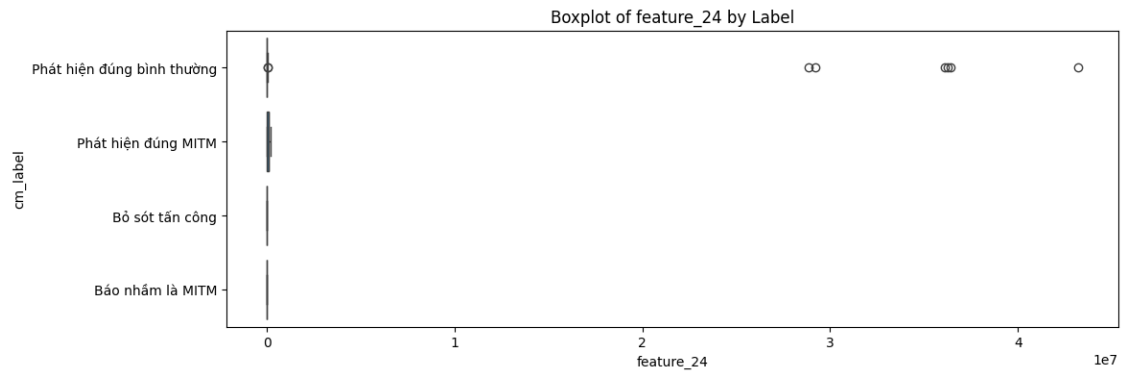
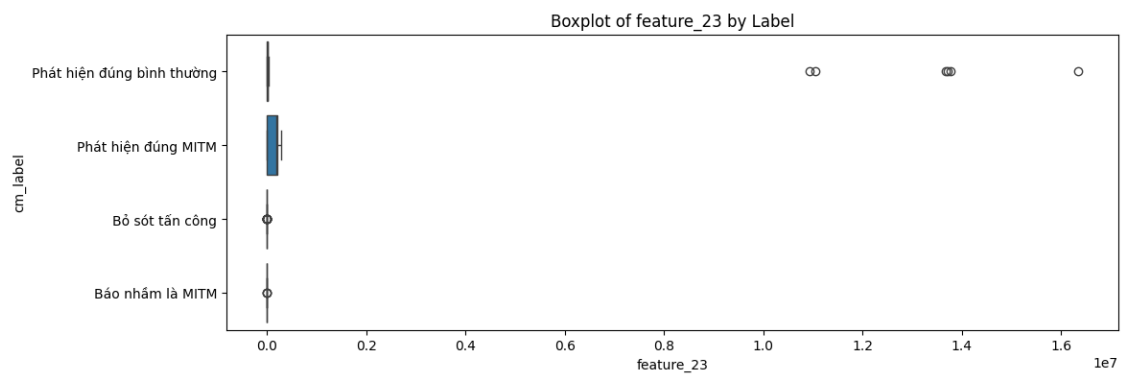
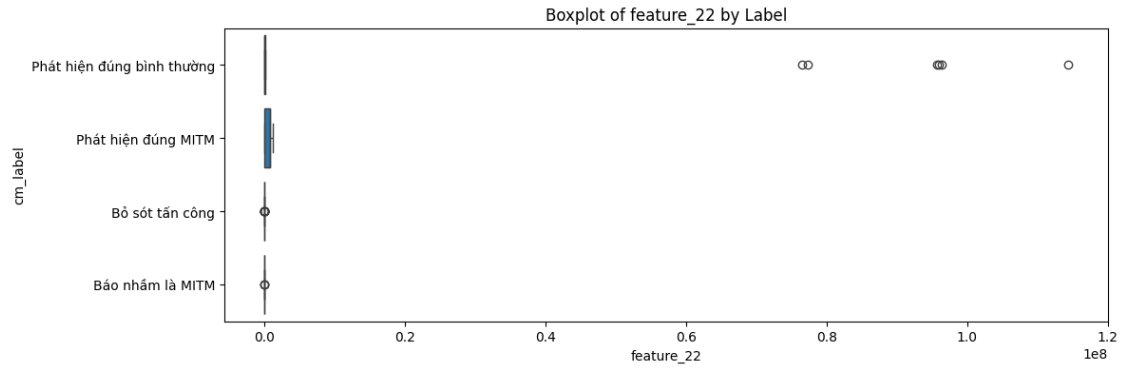


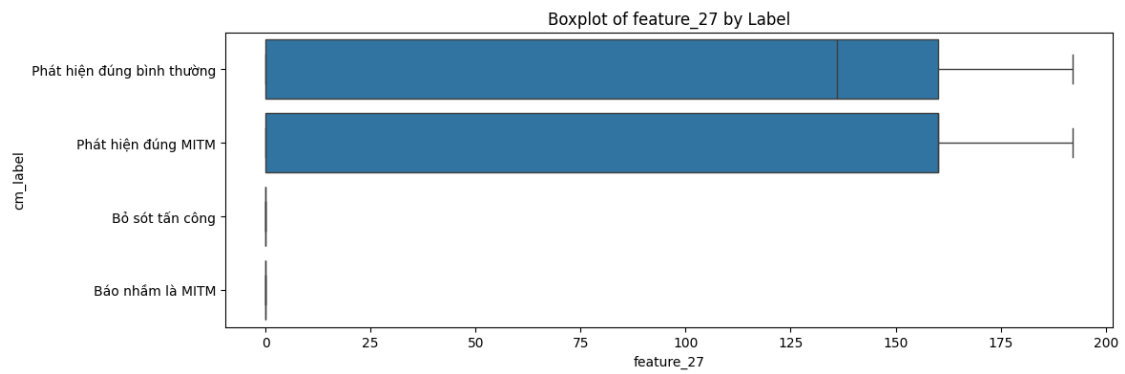
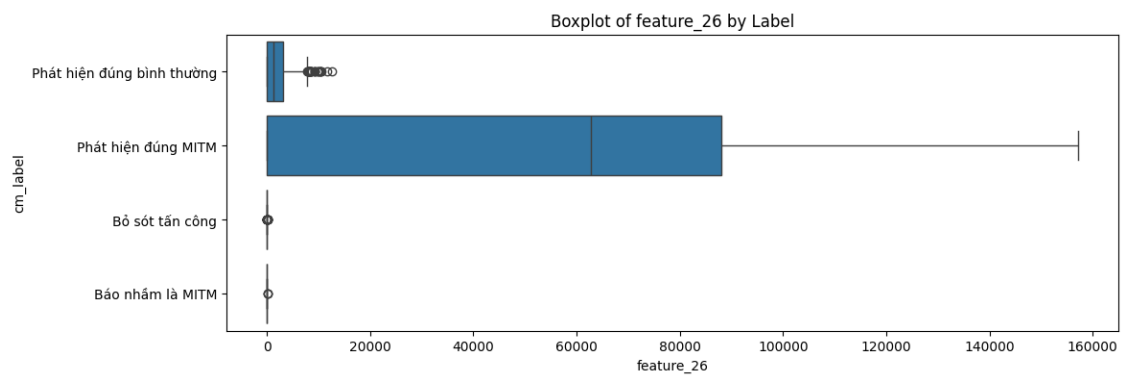
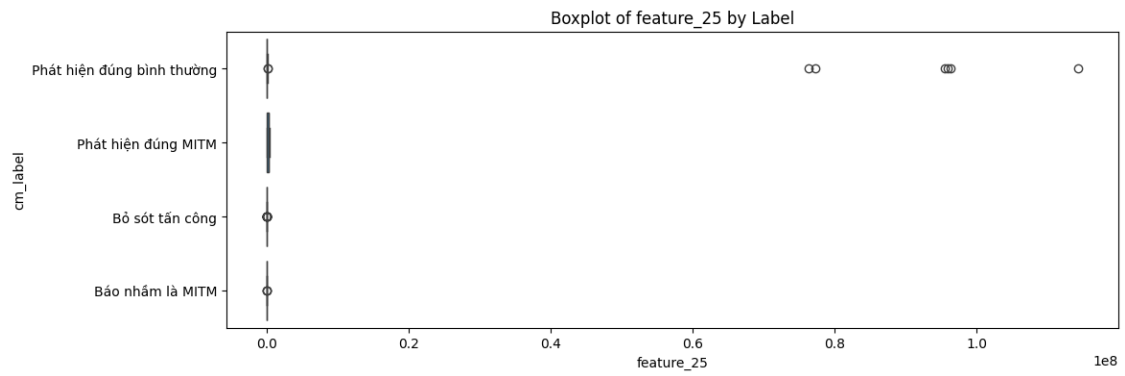


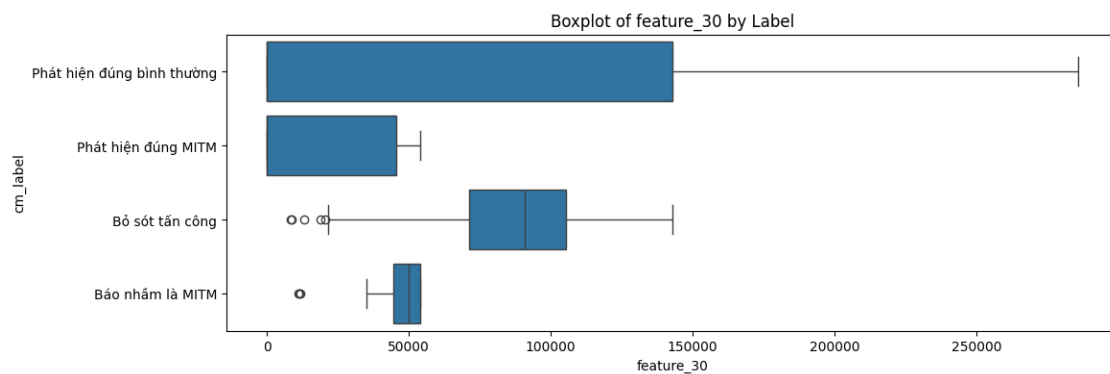
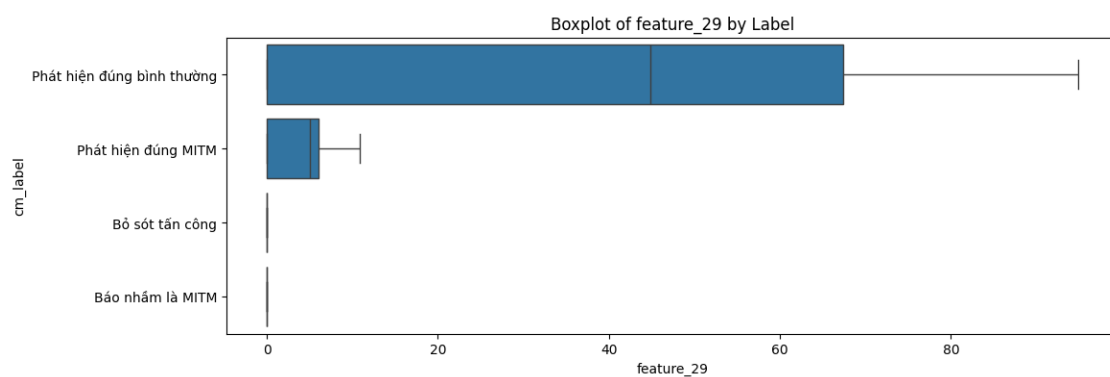
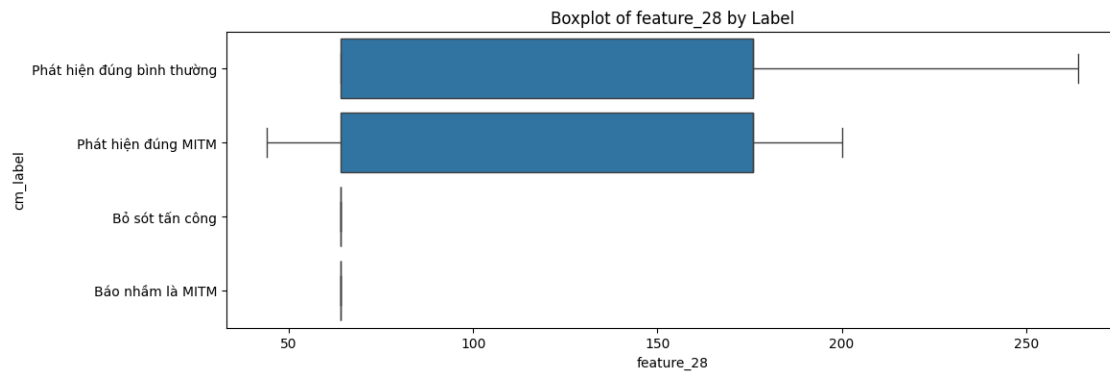


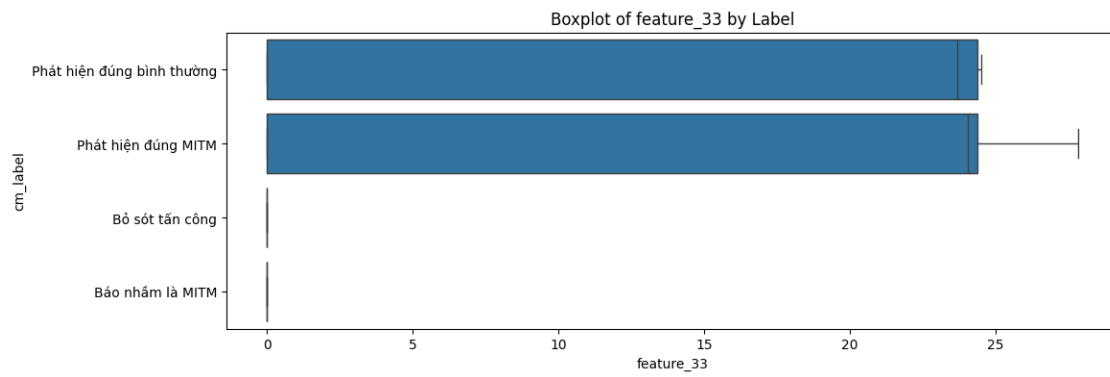
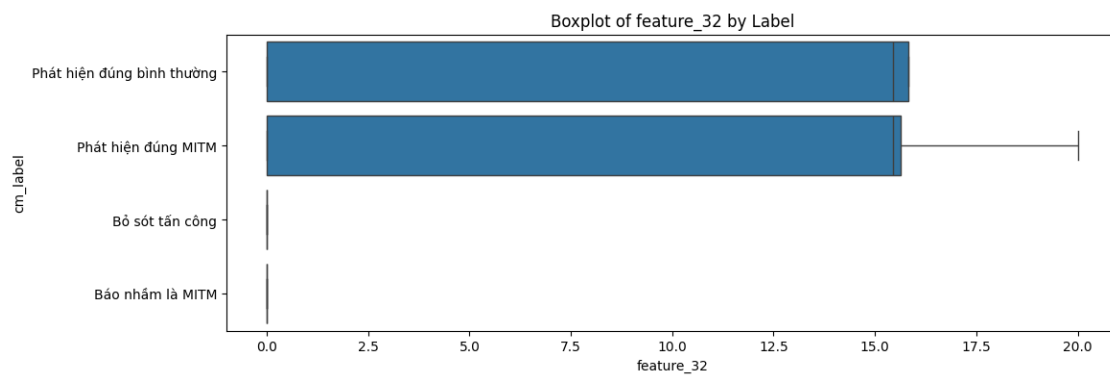
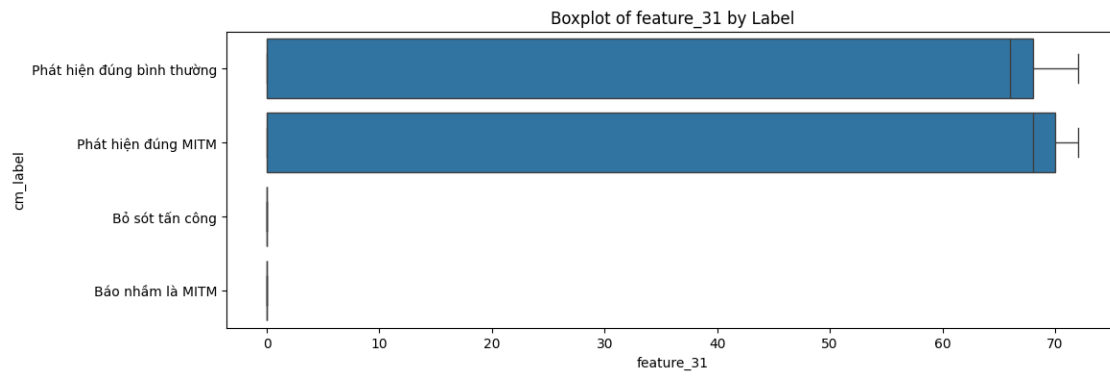


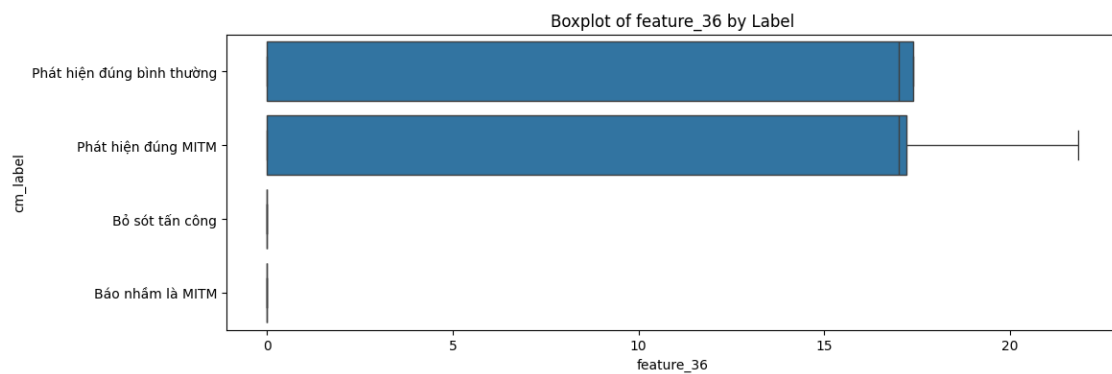
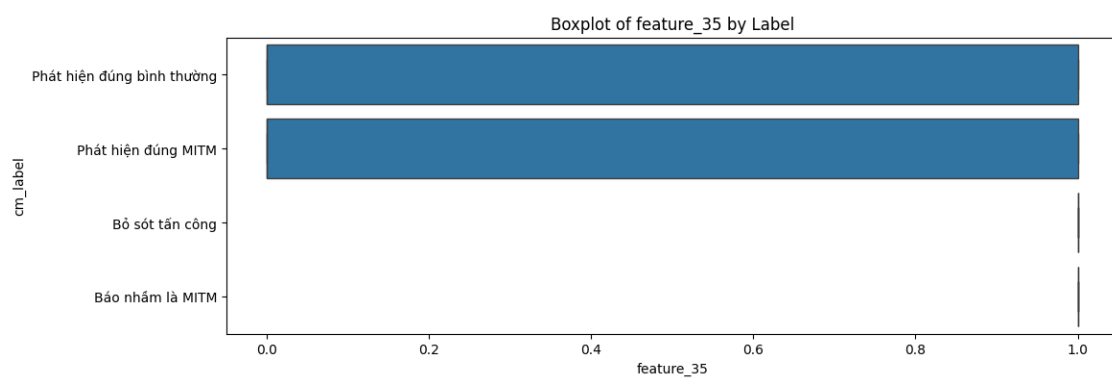
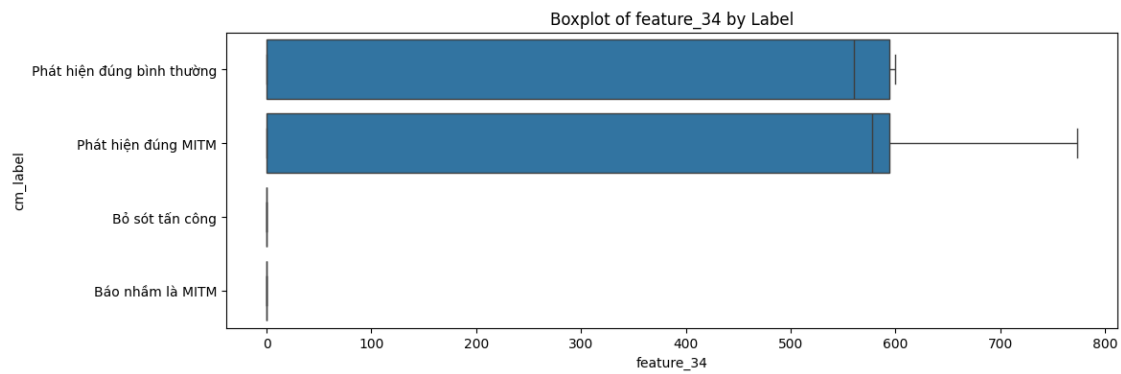


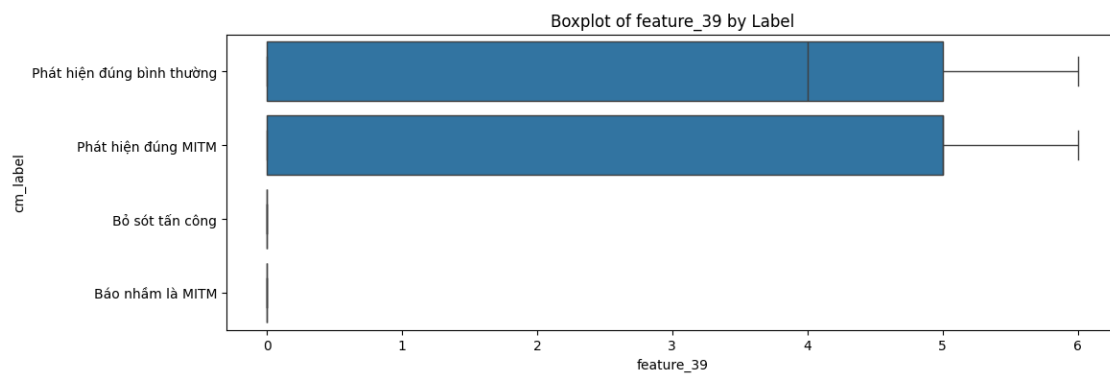
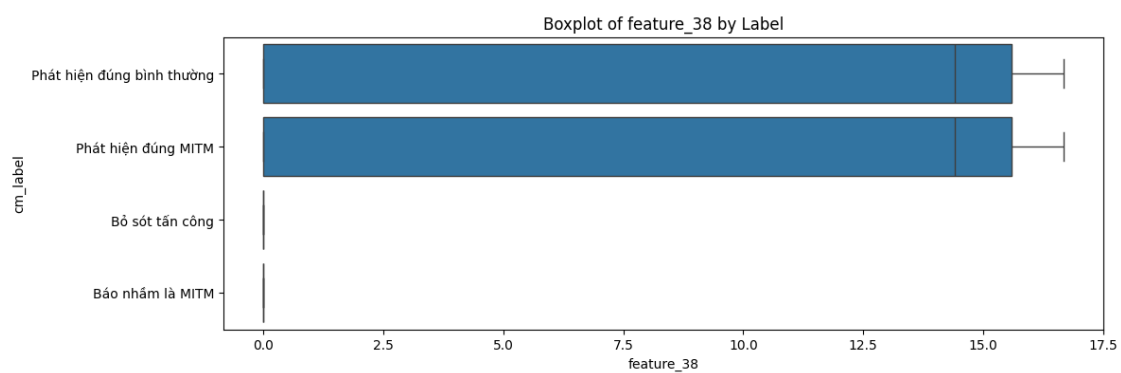
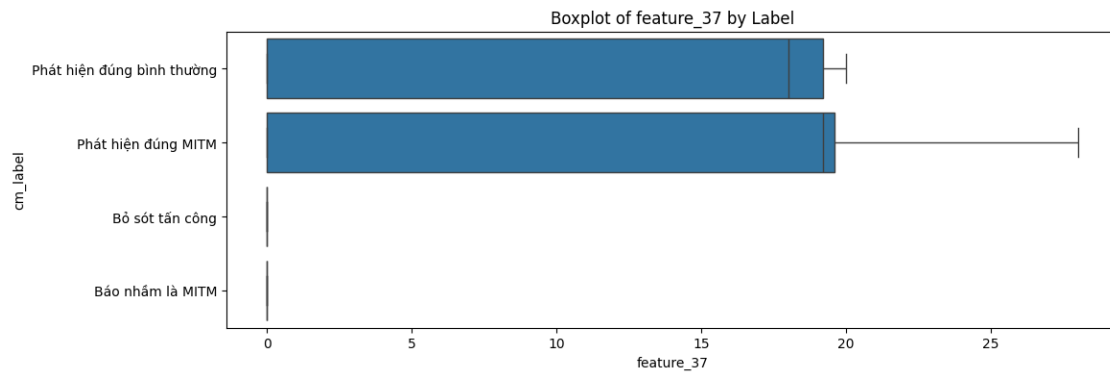


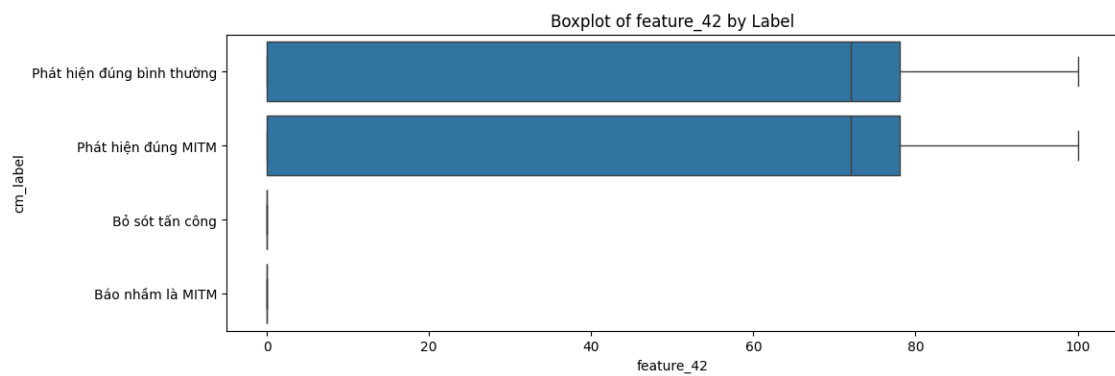
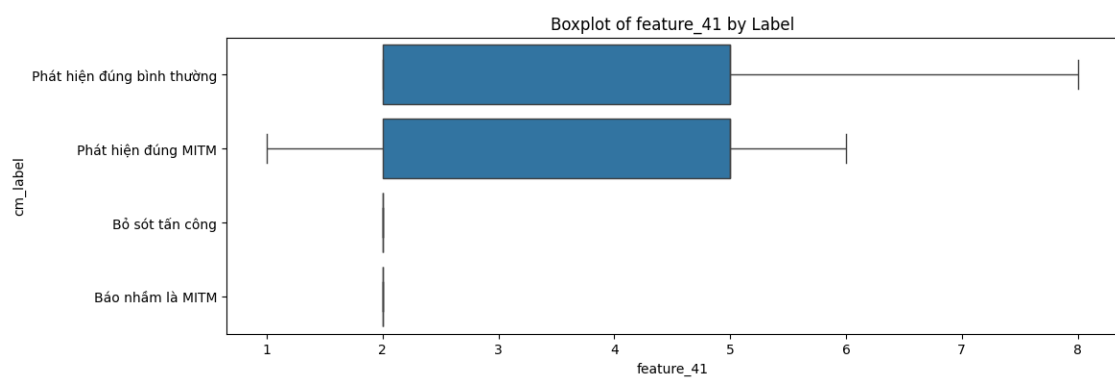
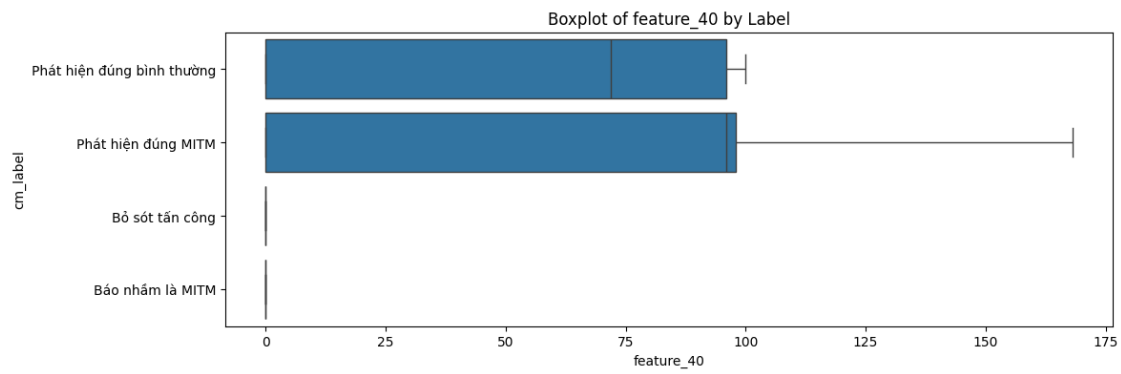


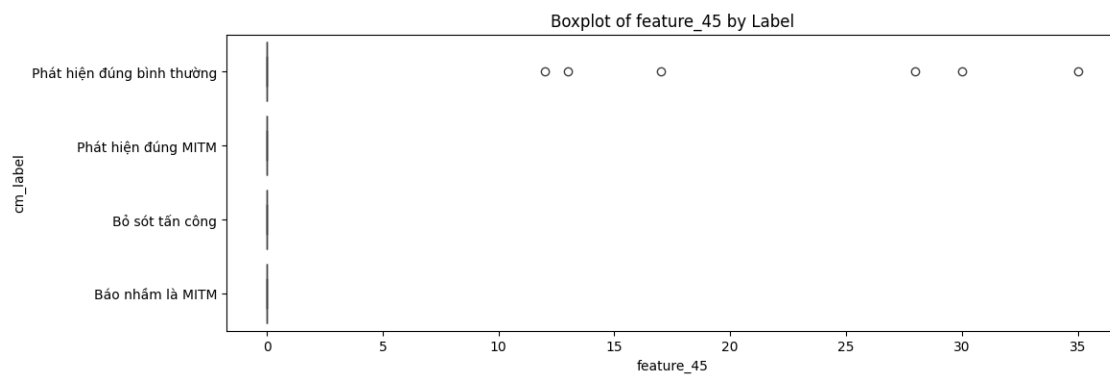
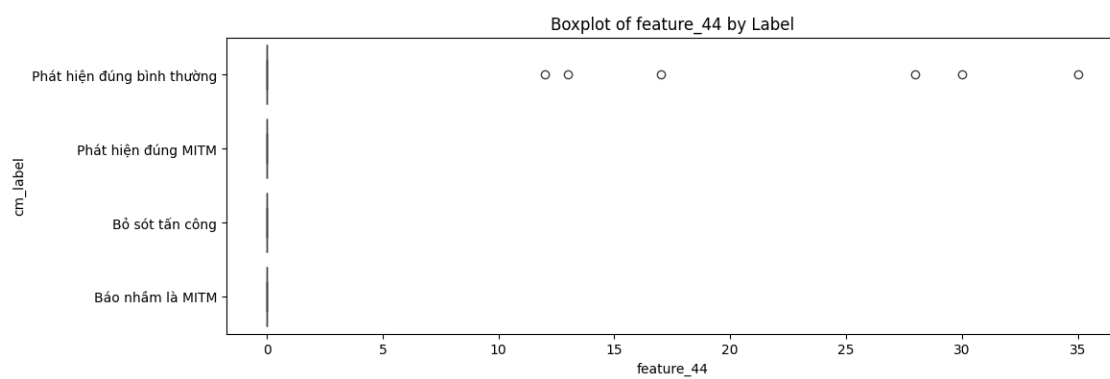
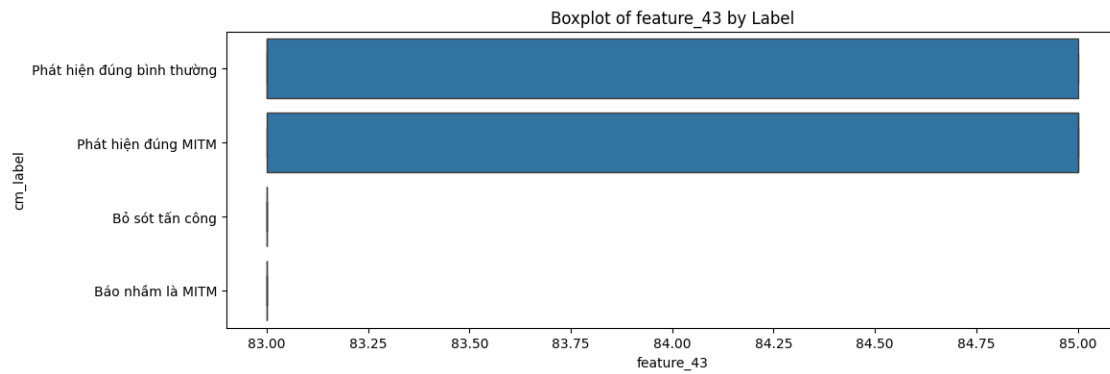


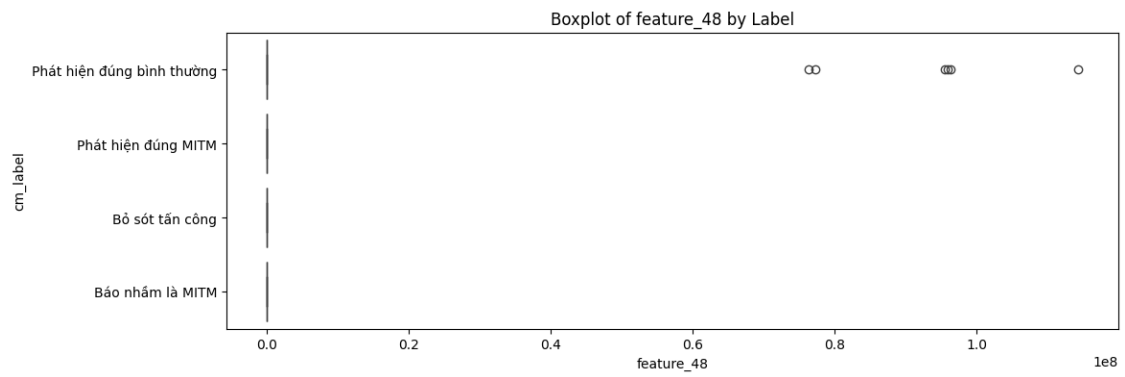
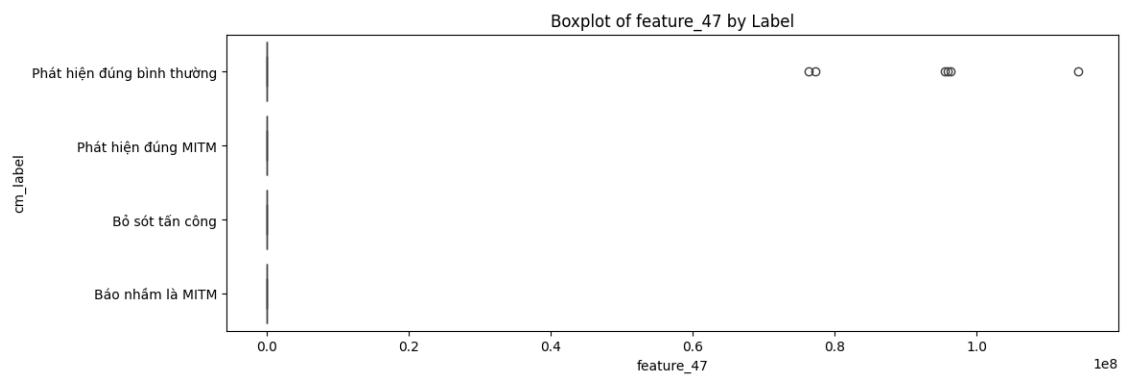
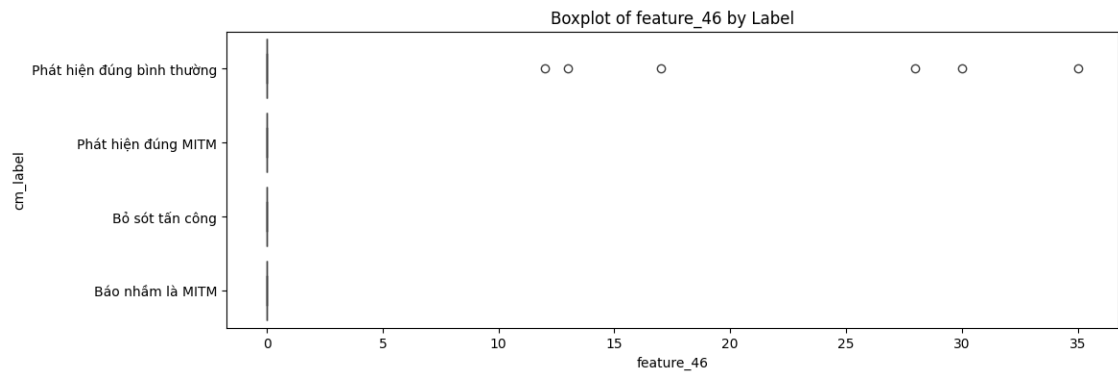


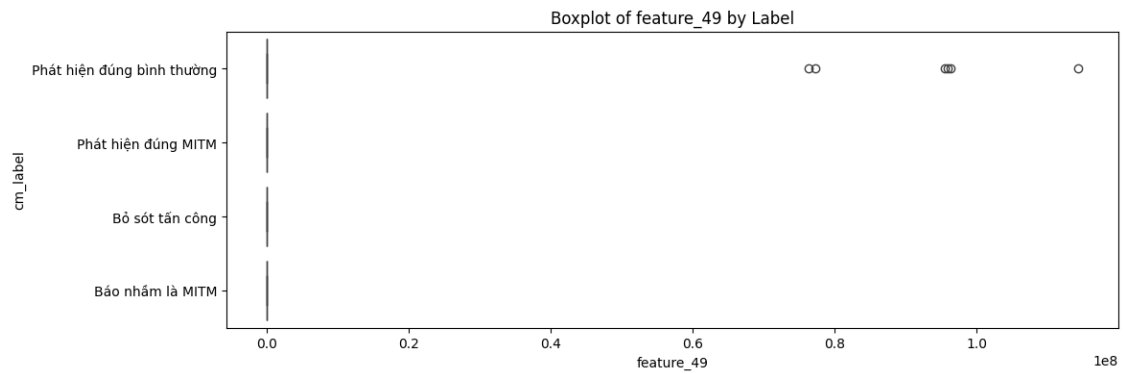












```
[23]: for col in df.select_dtypes(include='number').columns:
plt.figure(figsize=(12, 4))
sns.violinplot(y='cm_label', x=col, data=df)
plt.title(f'Violin plot of {col} by Label')
plt.show()
```

